# Findings of the WMT 2026 Shared Task on Model Compression: No Free Lunch at Extreme Compression

**Thamme Gowda**
Microsoft
thammegowda@microsoft.com

**Marco Gaido**
FBK
mgaido@fbk.eu

**Roman Grundkiewicz**
Microsoft
rogrundk@microsoft.com

**Matteo Negri**
FBK
negri@fbk.eu

## Abstract

We present the second edition of the WMT Model Compression(WMT26MC) shared task, which studies how large language models can be made efficient for machine translation under storage, memory, and latency constraints. WMT26MC offers a constrained track based on Gemma 3 12B and an unconstrained track for compressing models originally smaller than 20B parameters. Both tracks cover Czech→German, English→Chinese (Simplified), and English→Arabic (Egyptian). Participants submit complete runnable systems, which the we execute in a standardized environment with one NVIDIA H100 GPU. Evaluation jointly considers translation quality, model size on disk, GPU memory use, and decoding speed, and analyzes the resulting quality–size and quality–speed trade-offs. We received 17 submissions in the constrained track and 5 in the unconstrained track. The proposed systems span weight and activation quantization, structured and expert pruning, low-rank factorization, knowledge distillation, vocabulary and multimodal-component removal, mixed precision, and inference-time reranking. Evaluating with reference-free quality-estimation metrics on the blind test sets, we find that post-training quantization is the reliable lever: INT4 and FP8 systems shrink the constrained base to roughly a quarter of its on-disk size and run up to an order of magnitude faster while staying within quality-estimation noise of the uncompressed model, whereas aggressive low-rank and structural pruning collapse quality.

## 1 Introduction

Large language models (LLMs) have become strong general-purpose systems for machine translation(MT), but their broad capabilities come with substantial storage, memory, and computational requirements. These costs complicate deployment on commodity accelerators, local servers, mobile hardware, and other resource-constrained platforms. Compression methods such as quantization, pruning, low-rank factorization, and knowledge distillation can reduce these requirements, although aggressive compression may degrade translation quality or fail to yield a corresponding reduction in end-to-end latency (Zhu et al., 2024). A useful comparison therefore has to measure the resulting systems as deployable artifacts rather than considering parameter count or nominal precision alone.

The WMT 2025 Model Compression shared task introduced with objective to provide such a comparison for text translation (Gaido et al., 2025). It continued a line of WMT shared tasks on efficient translation (Birch et al., 2018; Hayashi et al., 2019; Heafield et al., 2020, 2021, 2022), complementing a parallel initiative in the speech translation field at IWSLT (Abdulmumin et al., 2025; Adelani et al., 2026). The second edition, WMT26MC, retains the central objective—reducing the footprint and cost of a general-purpose LLM while preserving machine-translation quality—but updates the benchmark to reflect both newer models and more realistic system deployment.

WMT26MC introduces several changes. The constrained track replaces Aya Expanse 8B with Gemma 3 12B, and the unconstrained track permits any source model whose original size is below 20B parameters. Both tracks use the same three language directions: Czech→German, English→Chinese (Simplified), and English→Arabic (Egyptian). Participants submit the complete runnable artifact rather than only model outputs, allowing organizers to measure quality, on-disk and GPU-memory footprints, and decoding speed under the same hardware and operating-system conditions. All systems are run by the organizers on an x86_64 host with Ubuntu 24.04 and a single NVIDIA

H100 GPU with up to 80 GB of VRAM.[1] The evaluation is designed to address three questions:

- How far can a general-purpose LLM be compressed while retaining its translation quality on the target language directions?
- Which combinations of compression and inference techniques provide the strongest quality–size, quality–memory, and quality–speed trade-offs?
- How do methods developed under a fixed-model, fixed-data constraint compare with unconstrained model selection and training?

This paper describes the submitted systems, the common evaluation pipeline, and report results.

## 2 Task Description

The task asks participants to reduce the deployable footprint of a general-purpose LLM while maintaining strong translation performance. Systems may target any subset of the three language directions, and rankings are computed separately for each track and direction.

### 2.1 Tracks

The **constrained track** establishes a common starting point by requiring all systems to be directly derived from `google/gemma-3-12b-it`.[2] Gemma 3 12B was selected for its permissive licensing terms and its balance of size and translation capability across the task languages. Any compression technique is permitted. However, the final model must remain a compressed derivative of the original model: for example, a distilled student is eligible only if its architecture is itself obtained from Gemma 3 12B rather than chosen independently.

The **unconstrained track** uses the same language directions and evaluation protocol but allows participants to choose any original model below 20B parameters. It also permits additional calibration and training data. By holding the evaluation environment fixed while relaxing model and data choice, this track tests whether model selection and task specialization offer better deployment trade-offs than compressing a common base model.

[1] https://www2.statmt.org/wmt26/model-compression.html

[2] https://huggingface.co/google/gemma-3-12b-it

### 2.2 Data

The task follows the WMT26 General MT task for its language setting, test data, and automatic translation-quality protocol. Constrained-track participants may use the public data released for the WMT26 campaign and test sets from previous WMT editions for calibration or fine-tuning. Unconstrained-track participants may use additional data. The official evaluation uses held-out WMT26 General MT data for the selected directions.

Evaluation inputs are exposed to submitted systems as UTF-8 text with one source record per line. Each system must return exactly one translation per input line. This line-oriented interface keeps data handling identical across model frameworks and prevents system-specific preprocessing from changing the number or order of evaluated examples.

### 2.3 Evaluation

Submissions are evaluated separately for each track and language along three dimensions:

1. **Translation quality**, assessed with reference-based and reference-free metrics. The official metrics are COMET, and MetricX.
2. **Model footprint**, including the complete artifact size on disk and peak GPU-memory use during inference.
3. **Decoding speed**, measured from the wall-clock time required to translate the evaluation set under controlled batch settings.

All systems run on an organizer-managed x86_64 machine with Ubuntu 24.04 and a single NVIDIA H100 GPU with 80 GB of VRAM. The same submission interface, input order, decoding workload, and hardware allocation are used for every system. We report the dimensions individually and visualize quality–size and quality–speed trade-offs with Pareto frontiers; a system is on a frontier when no other system is simultaneously better in both displayed dimensions.

The evaluation pipeline is released with the organizers' repository.[3] Before scoring, a sanity check installs each submission, invokes it on a small input, verifies that the output is non-empty, and enforces exact input/output line-count match. The main evaluator invokes the same entry point for all accepted systems and records the command, exit

[3] https://github.com/thammegowda/wmt26-model-compression

status, wall-clock time, CPU times, maximum host resident-set size, and timestamps for each run. Invalid or incomplete outputs are excluded from scoring. The current throughput protocol profiles nominal batch sizes 1, 16, 64, 256, and 512, with three measured runs per condition and warmup runs before measurement.

### 2.4 Submission

A valid entry consists of a submission form and a complete system that the organizers can execute without reconstructing the compression procedure. Each submission package contains source code, environment-installation instructions and dependencies, an inference entry point, and the compressed model artifact (or a stable link to it). Under the submission specification, packages provide `setup.sh`, `run.sh`, `requirements.txt`, and `README.md`. The evaluator calls `run.sh` with a language pair, batch size, input path, and output path; diagnostic output must not be mixed with translations.

Teams may submit multiple systems for a track and language. In that case they designate one system as PRIMARY, with the rest are treated as CONTRASTIVE. Participants also declare the language directions and maximum supported batch size for each submission.

## 3 Participants

We received participation from 13 teams; among which 10 teams participated in the constrained track, 4 teams target the unconstrained track; 1 team (`Pare4Bit`) entered both tracks. Many participants submitted multiple entries; in that case they were asked to select a single primary submission while the remaining submissions are treated as contrastive.

Table 1 reports the team names, number of submissions, track name, and a brief description of methods. The descriptions below consolidate the short paragraph responses in the submission form. They summarize systems as declared by participants; claims about quality, speed, or memory use are deferred to the common evaluation in Section 4, whose results are consolidated and presented in (Table 2).

### 3.1 alonso

The `alonso` entry is a calibration-free English→Chinese (Simplified) system that selectively applies BitsAndBytes NF4 with double quantization to the gate, up, and down projections of all 48 text-transformer MLP blocks. Attention projections, embeddings, normalization layers, the output head, and vision modules remain in BF16. This selective design trades a larger artifact for lower quantization exposure than uniform 4-bit compression.

### 3.2 arc/ilsp (Athena Research Center)

The `arc/ilsp` systems combine three training-free operations: removal of the vision tower, script-aware vocabulary pruning from roughly 262k to 196k entries, and GPTQ W4A16 quantization. The team selects full- or reduced- vocabulary INT4 primaries by language direction and includes an FP8 contrastive. For English→Arabic (Egyptian), a decoding contrastive generates multiple candidates and applies self-MBR selection using chrF consensus without changing model weights.

### 3.3 CometCut

`CometCut` compares uniform GPTQ and AWQ with a quality-aware mixed-precision scheme. For the latter, the team quantizes one decoder layer at a time, estimates its sensitivity using COMET on held-out data, and solves a knapsack problem that assigns layers to 16-, 8-, or 4-bit precision under a size budget. All variants use a shared multilingual calibration pool and the LLM-Compressor/vLLM stack.

### 3.4 Dragon1006 (IIIT)

`Dragon1006`'s unconstrained English→Chinese (Simplified) system starts from Aya Expanse 8B, reduces its depth from 32 to 28 layers, and restores translation behavior with QLoRA on a 36k-example subset of WMT19. The adapter is merged into the model to remove PEFT routing overhead, and weights are quantized to 8-bit at inference time with BitsAndBytes. Retrieved after the initial inventory snapshot, the int8 artifact is evaluated here on English→Chinese (Simplified) (Table 2); note that its inference wrapper caps generation at 150 new tokens, which truncates the longer paragraphs of the WMT26 set and depresses its measured quality.

### 3.5 ESTS (UCLA)

`ESTS` builds unconstrained translation specialists from the MXFP4-quantized GPT-OSS-20B mixture-of-experts model. It estimates expert

| ID | Team | Track | Directions | # Var. | Main techniques | Details |
|---|---|---|---|---|---|---|
| Al | alonso | C | en–zh | 1 | Selective MLP-only NF4 quantization | §3.1 |
| Ar | arc/ilsp | C | all | 4 | Vision/vocabulary pruning, GPTQ/FP8, self-MBR | §3.2 |
| C | CometCut | C | all | 3 | GPTQ, AWQ, COMET-guided mixed precision | §3.3 |
| D | Dragon1006 | U | en–zh | 1 | Layer pruning, LoRA recovery, runtime INT8 | §3.4 |
| E | ESTS | U | en–zh, en–ar | 6 | MoE expert pruning, MXFP4, supervised recovery | §3.5 |
| F | FBK@ModelCompression | C | cs–de | 2 | SmoothQuant, GPTQ, W8A8 activation quantization | §3.6 |
| P | Pare4Bit | C/U | all | 3 | Vision/depth pruning, GPTQ INT4, model selection | §3.7 |
| S | SLICERS | C | en–zh | 1 | Layer/FFN pruning, NF4, QLoRA | §3.8 |
| Ta | TahomaMT | C | all | 4 | Vision/vocabulary pruning, FP8/INT4, QE reranking | §3.9 |
| Ti | TildeOpen-wmt | U | cs–de | 5 | Distillation, NVFP4/FP8 quantization, GRPO | §3.10 |
| Tn | Tiny Titans | C | cs–de | 3 | GPTQ, activation-aware SVD, distillation | §3.11 |
| TM | TMU-onono | C | cs–de | 2 | DiBA factorization and diagonal-only recovery | §3.12 |
| V | VIC | C | all | 21 | Structured pruning, vocabulary pruning, AWQ | §3.13 |

Table 1: Registered WMT26 participants and their 56 submitted participant variants across all 13 teams (15 pending download). The ID column defines the participant labels used throughout the tables and figures. C and U denote constrained and unconstrained tracks; "all" denotes cs–de, en–zh, and en–ar. Variant counts describe artifacts, not submission-form rows.

importance from routing on task-proxy generations and allocates retained capacity across layers using the Jensen–Shannon divergence between source- and target-side routing distributions. Experts are physically removed, after which the model is recovered with full-parameter supervised training on GPT-generated pseudo-parallel data. Separate English–Chinese and English–Egyptian-Arabic families vary the number of retained experts.

### 3.6 FBK@ModelCompression

FBK submits two W8A8 Czech→German systems. Both use GPTQ for 8-bit weights and round-to-nearest quantization for activations, executing the matrix operations directly in 8-bit precision through the compressed-tensors/vLLM stack. One variant first applies SmoothQuant to move part of the activation quantization difficulty into the weights; the other omits this smoothing step.

### 3.7 Pare4Bit

For the constrained track, Pare4Bit rebuilds Gemma 3 12B as a text-only model without the vision tower and applies group-wise GPTQ W4A16 while retaining tied embeddings in BF16. A contrastive additionally drops four transformer layers, uses LoRA knowledge-distillation recovery, and then applies GPTQ. Its unconstrained entry treats model selection as part of compression, choosing a Gemma 4 12B quantization-aware-trained INT4 checkpoint and serving it through vLLM.

### 3.8 SLICERS (IIIT Hyderabad)

SLICERS targets English→Chinese (Simplified) with a combination of structural and precision compression. Fisher-information estimates guide removal of eight transformer layers and 30% of intermediate feed-forward neurons. The structurally pruned model is then quantized to 4-bit NF4 and adapted with QLoRA on translation data.

### 3.9 TahomaMT (Microsoft)

TahomaMT removes the unused SigLIP vision tower and multimodal projector and specializes the tokenizer to the Latin, Arabic, and Han scripts used in the task, reducing its vocabulary from 262,208 to 192,347 entries with byte fallback. Two calibration-free variants use either FP8 weights and per-token FP8 activations (W8A8) or group-wise asymmetric INT4 weights with FP8 activations (W4A8). Contrastive best-of-four systems sample four candidates and select one with a bundled 4-bit Cometoid quality-estimation model. All variants use the team's C++/CUDA Tahoma runtime.

### 3.10 TildeOpen-wmt (Tilde)

Tilde's unconstrained submission starts from a task-adapted TildeOpen 15B Czech–German model, itself distilled from a 30B model and further trained with supervised fine-tuning and GRPO. The submission compares GPTQ with NVFP4 weights against FP8 weights and dynamic FP8 activations. A second compression axis distills the 15B model to an 8B student, and the team also com-

bines this student with each quantization scheme.

### 3.11 Tiny Titans (MBZUAI)

`Tiny Titans` submits three Czech→German systems. The first applies group-wise 4-bit GPTQ using a 1,024-example calibration set. The second factorizes selected weight matrices with activation-aware SVD and then performs task-aware distillation from the original Gemma model, using stored top-32 teacher probabilities. The third combines this low-rank/distillation pipeline with 4-bit GPTQ of the resulting factors.

### 3.12 TMU-onono (Tokyo Metropolitan University)

`TMU-onono` applies DiBA (Diagonal and Binary Matrix Approximation), replacing dense linear and tied embedding/output weights with interleaved diagonal scalings and binary mixing factors. The binary factors remain fixed while the diagonal factors are adapted on teacher translations of the WMT26 Czech→German training data, a recovery procedure the team calls DiBARD. Its primary system uses a direct Triton backend, while a contrastive system caches unpacked factors at runtime.

### 3.13 VIC (Vicomtech)

`VIC` removes Gemma's vision component and learns differentiable masks over feed-forward channels and attention groups while keeping the model weights fixed. The masks are materialized as structured pruning at several target ratios; selected models are then recovered by supervised fine-tuning on multilingual parallel data. The submission family also explores language-specific vocabulary pruning and applies 4-bit AWQ after pruning, with inference served through vLLM.

Quantization is the most common component, but it is frequently combined with structural specialization: teams remove vision modules unused by text-only MT, prune vocabularies or transformer components, factorize dense matrices, or distill pruned students. The unconstrained entries broaden model choice to AyaExpanse, Gemma 4, GPT-OSS, and TildeOpen models. Several entries also optimize the inference system itself through custom kernels, continuous batching, or quality-estimation-based selection. Final evaluated-system counts will be reported after artifact validation.

## 4 Results

We evaluate all submissions in a single standardized environment and report the three task dimensions separately. Because the WMT26 test sets are blind at the time of writing, the quality numbers here are *reference-free* quality-estimation (QE) scores; reference-based COMET/MetricX and the LLM-as-a-judge protocol will be added once the references are released. Unless noted otherwise, quality is COMET-Kiwi-XXL (Rei et al., 2023), cross-checked against MetricX-XXL-QE (Juraska et al., 2024). The constrained track compresses a common base (Gemma 3 12B, 22.7 GiB on disk); the unconstrained track uses heterogeneous bases, so its systems are compared against the constrained base only as a reference line, not on equal footing. Throughout, the uncompressed base and the two `bitsandbytes` organizer baselines (`bnb-q4`, `bnb-q8`) anchor the plots. The full per-system results are consolidated in Table 2, the central table of this report; the figures that follow break out each dimension.

### 4.1 Translation Quality

Table 2's results for quality dimension indicate the following: post-training *quantization* preserves quality, whereas aggressive *structural* compression does not. Weight-only and weight–activation quantization (INT4/W4A16, FP8/W8A8, INT8) land within QE noise of the uncompressed base, while low-rank factorization (`Tiny Titans`' SVD, COMET-Kiwi-XXL 25–32 for Czech→German), binary/diagonal approximation (`TMU-onono`'s DiBA, 27–28), and 50% FFN pruning without sufficient recovery (Vicomtech, down to 49 for English→Arabic (Egyptian)) collapse quality. The failures reflect capacity removal that the submitted recovery (distillation or light fine-tuning) did not restore. Some models using calibrated data improve on CometKiwiXXL over baselines but worsen MetricX24 scores.

### 4.2 Model Size

Figure 1 shows the quality–size trade-offs separately for each language direction under both QE metrics. Both tracks appear in every panel; Table 2 reports every evaluated system, grouped by participant, with per-direction quality and speed ranks. The headline result is that the base model can be shrunk to roughly a quarter of its on-disk size at essentially no quality cost: the

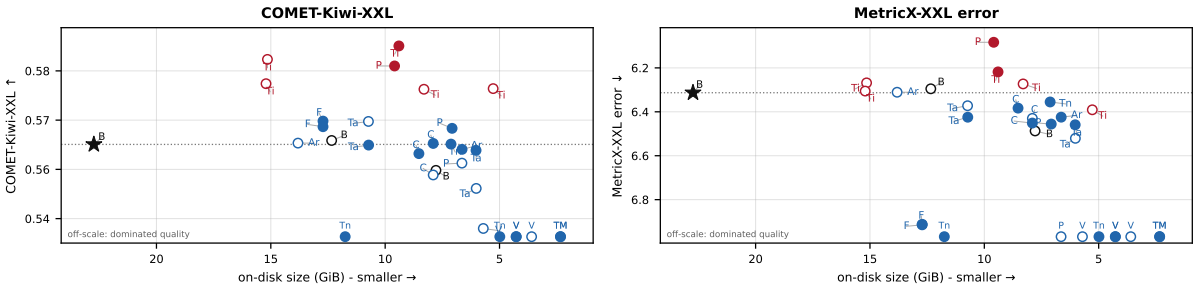
**Constrained track (compress Gemma-3-12B)**

| System | | Data | Size↓ | c/s↑ | cs→de CK↑ | cs→de MX↓ | en→zh CK↑ | en→zh MX↓ | en→ar CK↑ | en→ar MX↓ |
|---|---|---|---|---|---|---|---|---|---|---|
| B/base★ | HF | — | 20 22.74 | 22 735 | 4 56.26 | 3 6.31 | 3 76.95 | 5 2.77 | 6 61.51 | 3 5.36 |
| B/bnb-q8★ | HF+BnB | — | 16 12.34 | 24 393 | 3 56.38 | 2 6.30 | 5 76.84 | 7 2.80 | 8 61.05 | 5 5.49 |
| B/bnb-q4★ | HF+BnB | — | 10 7.78 | 21 800 | 9 55.47 | 12 6.49 | 7 76.65 | 8 2.81 | 11 60.66 | 8 5.62 |
| ● **Al/mlp-nf4** | HF+BnB | — | 14 10.99 | 23 604 | 3 56.38 | 1 6.23 | 6 76.69 | 6 2.78 | – | – |
| ● **Ar/int4** | vLLM | cal | 9 7.08 | 15 2816 | 6 56.10 | 13 6.52 | 8 76.46 | 10 2.85 | – | – |
| ● **Ar/vocaball-int4** | vLLM | cal | 8 6.64 | 10 3543 | 6 56.11 | 8 6.42 | – | – | 14 60.10 | 10 5.67 |
| ○ Ar/fp8 | vLLM | cal | 18 13.81 | 11 3414 | 4 56.30 | 3 6.31 | 4 76.91 | 3 2.75 | 10 60.84 | 6 5.51 |
| ○ Ar/en-ar-mbr | vLLM | cal | 8 6.64 | 22 720 | 6 56.12 | 7 6.40 | – | – | 7 61.42 | 7 5.54 |
| ● **C/mixed** | vLLM | cal | 12 8.53 | 14 2899 | 7 55.98 | 6 6.38 | 3 76.98 | 3 2.75 | 13 60.19 | 4 5.46 |
| ● **C/gptq** | vLLM | cal | 11 7.90 | 14 2885 | 4 56.29 | 10 6.45 | 7 76.55 | 4 2.76 | 12 60.58 | 8 5.62 |
| ○ C/awq | vLLM | cal | 11 7.90 | 13 3031 | 10 55.33 | 9 6.43 | 10 76.02 | 9 2.82 | 9 60.99 | 10 5.67 |
| ● **F/gptq-rtn** | vLLM | cal | 17 12.72 | 9 3997 | 1 56.97 | 14 6.91 | – | – | – | – |
| ● **F/sq-gptq** | vLLM | cal | 17 12.72 | 8 4199 | 2 56.80 | 14 6.91 | – | – | – | – |
| ● **P/int4-gptq** | vLLM | cal | 9 7.07 | 19 1671 | 2 56.75 | 11 6.46 | 9 76.44 | 6 2.78 | 14 60.14 | 11 5.71 |
| ○ P/pruned4-int4 | vLLM | FT | 8 6.64 | 18 1785 | 8 55.69 | 15 7.41 | 11 75.83 | 11 2.97 | 5 61.99 | 12 5.73 |
| ● **S/pruned-nf4** | HF+BnB | FT | 19 15.44 | 23 565 | 19 10.54 | 25 22.38 | 16 56.88 | 13 5.62 | – | – |
| ● **Ta/fp8** | Tahoma | — | 13 10.72 | 2 5591 | 5 56.24 | 8 6.42 | 6 76.74 | 6 2.78 | 9 60.99 | 5 5.49 |
| ● **Ta/int4-bok4** | Tahoma | — | 7 6.03 | 17 2210 | 6 56.08 | 11 6.46 | 2 77.06 | 2 2.69 | 2 64.31 | 1 5.08 |
| ○ Ta/fp8-bok4 | Tahoma | — | 13 10.73 | 16 2408 | 1 56.96 | 5 6.37 | 1 77.49 | 1 2.66 | 3 63.12 | 2 5.29 |
| ○ Ta/int4 | Tahoma | — | 7 6.02 | 7 4890 | 11 54.92 | 13 6.52 | 12 75.74 | 9 2.82 | 4 62.11 | 4 5.46 |
| ● **Tn/svd-q512-f1620** | HF | FT | 15 11.75 | 25 17 | 14 31.53 | 23 19.78 | – | – | – | – |
| ● **Tn/gptq-int4** | HF+GPTQ | cal | 9 7.12 | 20 937 | 4 56.27 | 4 6.36 | – | – | – | – |
| ● **Tn/svd-q512-f1620-gptq4** | HF | FT | 5 4.99 | 20 877 | 18 24.31 | 24 21.68 | – | – | – | – |
| ● **TM/diba-triton** | DiBA/Triton | FT | 1 2.33 | 24 445 | 15 27.87 | 22 18.85 | – | – | – | – |
| ○ TM/diba-cached | DiBA/PT | FT | 1 2.33 | 12 3160 | 16 27.40 | 21 18.80 | – | – | – | – |
| ● **V/r50-ffn-sft-awq4** | vLLM | FT | 4 4.27 | 3 5475 | 13 44.33 | 18 14.38 | 15 58.04 | 14 5.70 | 15 58.30 | 13 7.67 |
| ○ V/r25-ffn-awq4 | vLLM | FT | 6 5.71 | 6 5015 | 12 53.70 | 16 10.37 | 13 72.48 | 12 3.67 | 1 70.69† | 9 5.63 |
| ○ V/r50-ffn-awq4 | vLLM | FT | 4 4.27 | 4 5317 | 17 26.47 | 20 16.76 | 17 55.57 | 15 5.74 | 16 48.85 | 15 10.10 |
| ○ V/r50-ffn-sft-awq4-en-zh | vLLM | FT | 3 3.74 | 5 5111 | 13 44.28 | 19 14.59 | 14 58.05 | 16 5.77 | – | – |
| ○ V/r50-ffn-sft-awq4-en-ar | vLLM | FT | 3 3.66 | 6 5046 | 13 44.26 | 17 14.23 | – | – | 15 58.26 | 14 7.69 |
| ○ V/r50-ffn-sft-awq4-cs-de | vLLM | FT | 2 3.60 | 1 6477 | 13 44.28 | 17 14.23 | – | – | – | – |

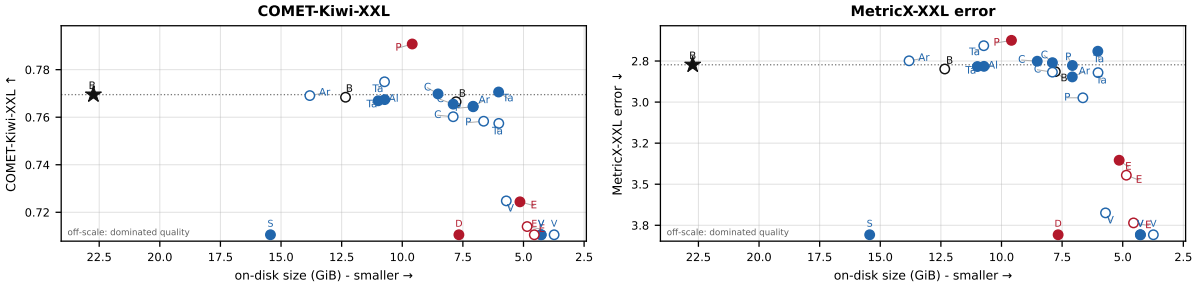
**Unconstrained track (heterogeneous base < 20B params)**

| System | | Data | Size↓ | c/s↑ | cs→de CK↑ | cs→de MX↓ | en→zh CK↑ | en→zh MX↓ | en→ar CK↑ | en→ar MX↓ |
|---|---|---|---|---|---|---|---|---|---|---|
| ● **D/aya-8b-int8** | HF+BnB | cal | 7 7.66 | 6 590 | – | – | 5 66.02 | 5 4.52 | – | – |
| ● **E/arz-k22** | vLLM | FT | 6 6.33 | 7 503 | – | – | – | – | 2 62.02 | 2 5.84 |
| ● **E/zho-k26** | vLLM | FT | 3 5.15 | 8 319 | – | – | 2 72.44 | 2 3.35 | – | – |
| ○ E/arz-k24 | vLLM | FT | 5 5.74 | 10 100 | – | – | – | – | 3 60.81 | 3 6.05 |
| ○ E/arz-k26 | vLLM | FT | 3 5.15 | 10 94 | – | – | – | – | 4 56.78 | 4 7.01 |
| ○ E/zho-k27 | vLLM | FT | 2 4.85 | 8 310 | – | – | 3 71.40 | 3 3.45 | – | – |
| ○ E/zho-k28 | vLLM | FT | 1 4.55 | 9 249 | – | – | 4 68.90 | 4 3.74 | – | – |
| ● **P/gemma4-int4** | vLLM | — | 10 9.59 | 5 1070 | 3 58.65 | 1 6.08 | 1 79.08 | 1 2.62 | 1 68.22 | 1 4.49 |
| ● **Ti/15b-nvfp4** | vLLM | FT | 9 9.40 | 4 1407 | 1 59.26 | 2 6.22 | – | – | – | – |
| ○ Ti/8b-distill | vLLM | FT | 12 15.21 | 1 3552 | 4 58.11 | 4 6.31 | – | – | – | – |
| ○ Ti/15b-fp8 | vLLM | FT | 11 15.15 | 4 1448 | 2 58.85 | 3 6.27 | – | – | – | – |
| ○ Ti/8b-distill-fp8 | vLLM | FT | 8 8.30 | 2 2891 | 6 57.94 | 3 6.27 | – | – | – | – |
| ○ Ti/8b-distill-nvfp4 | vLLM | FT | 4 5.28 | 3 2228 | 5 57.96 | 5 6.39 | – | – | – | – |

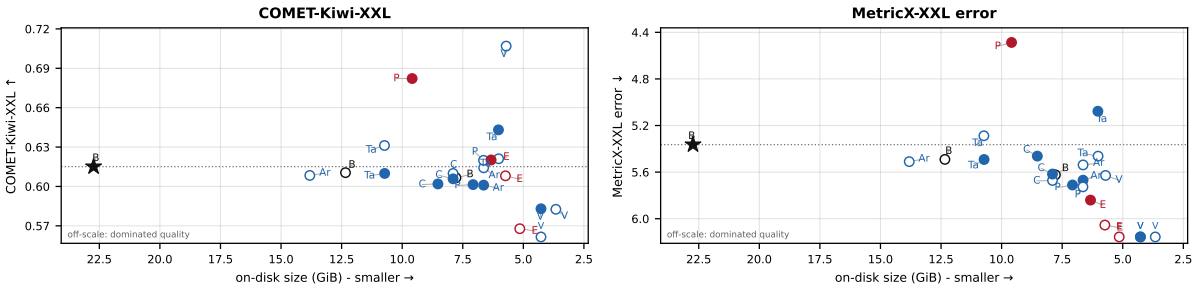
Table 2: **Model compression task results along size, speed, quality dimensions.** All systems are categorized by track (constrained=top and unconstrained=bottom), and grouped by participant ID. Primary submissions are shown first (solid circles, bold text), followed by contrastive submissions (hallow circles). The serving runtime is shown in the grey tag; the **Data** column marks whether the method is data-free (**—**, on-the-fly quantization), calibration-only (**cal**), or uses fine-tuning/distillation/healing (**FT**). Metric columns: **Size** (GiB), **c/s** mean end-to-end throughput (characters/second), **CK** COMET-Kiwi-XXL QE on a 0–100 scale (↑), and **MX** MetricX-XXL error (↓). Each cell is shaded by the system's rank *within its own block and column*, from green (best) through yellow to red (worst), with the rank number in the top-left corner; near-identical scores are grouped to share a rank and colour (grouping thresholds: Size 0.1 GiB, speed 100 chars/s, CK 0.1, MX 0.01). ↑/↓ in the header gives the "better" direction; the two tracks are ranked independently. Participant IDs follow Table 1. ⋆: organizer baseline.




(a) Czech→German quality vs. on-disk size.




(b) English→Chinese (Simplified) quality vs. on-disk size.




(c) English→Arabic (Egyptian) quality vs. on-disk size.

Figure 1: Quality vs. on-disk size for the three translation directions. Blue and red denote constrained and unconstrained systems; filled and hollow markers denote primary and contrastive submissions. Labels are participant IDs from Table 1; the star is the uncompressed baseline.

smallest near-lossless systems are INT4 quantizations at 6.0–6.6 GiB (26–29% of the base)—`arc/ilsp/vocaball-int4` for Czech→German, `TahomaMT/int4-bok4` for English→Chinese (Simplified), and `TahomaMT/int4` for English→Arabic (Egyptian)—all within ±0.01 COMET-Kiwi of the uncompressed base. FP8 (W8A8) is essentially lossless at about half the size (10.7 GiB), and several INT8 and mixed-precision systems sit just below the base. Vision-tower removal and script-specific vocabulary pruning contribute additional, quality-neutral savings on top of quantization. Beyond quantization, the frontier degrades sharply: the very smallest artifacts (2.3–4.6 GiB) are the structural-compression systems whose quality has collapsed, so they are Pareto-optimal only in the trivial sense of being small.

In the unconstrained track (Figure 1), *model selection acts as compression*. Pare4Bit's INT4 Gemma-4-12B exceeds the Gemma-3 base on all three directions (+0.02 to +0.07 COMET-Kiwi) at 9.6 GiB, and MetricX confirms the gain (e.g. English→Arabic (Egyptian) 4.49 vs. 5.36). `Tilde`'s distilled 8B student, quantized to NVFP4, stays above the Gemma-3 base for Czech→German at only 5.3 GiB, and UCLA's expert-pruned GPT-OSS-20B (`ESTS`) reaches 4.5–5.2 GiB by trading measured quality for size. These systems show that a well-chosen, newer sub-20B base—compressed—can dominate a compressed older base.

### 4.3 Inference Speed

Table 2 also reports end-to-end throughput averaged across three language pairs. Compres-

sion buys large speedups: the uncompressed base runs at 636–881 chars/s end-to-end, while near-lossless INT4/FP8 systems reach up to $\sim$ 6000 chars/s—about 7–10$\times$ faster at matched quality. `TahomaMT/fp8` occupies the quality–speed corner in every direction (6.2–9.5$\times$ the base throughput at near-baseline quality).

Crucially, throughput is governed as much by the serving runtime as by the precision. Optimized runtimes (Tahoma's C++/CUDA engine, vLLM continuous batching) dominate, whereas a naive `bitsandbytes` INT8 baseline is pathologically slow: on the Czech→German single-stream latency probe it takes 12,234 s ($\approx$3.4 h) versus 83 s for the fastest system, a $\sim 150\times$ spread that is entirely a runtime effect. Reducing precision is thus necessary but not sufficient for speed—the deployment stack must exploit it. We report throughput scaling with batch sizes {1, 16, 64, and 256} only for Czech→German (the direction for which most submissions opted-in) in Table 3.[4]

**Detailed Inference Speed** Figure 2 shows how net throughput scales with batch size on the Czech→German speed subset, and Table 3 lists per-system net rates at each batch. Throughput grows almost linearly with batch size until a system reaches its declared maximum batch or exhausts memory: the fastest systems are Vicomtech's FFN-pruned AWQ variants ($\sim$ 15k chars/s at batch 64) and `TahomaMT`'s FP8/INT4 on its custom engine ($\sim$ 10–13k at batch 256), an order of magnitude above the uncompressed and `bitsandbytes` baselines. Rates are load-subtracted (steady-state); cells where model load dominated the wall time are omitted as unreliable.

## 5 Conclusion and Future Directions

The second edition of WMT Model Compression shared task attracted participants with a variety of methods, which offer useful data points to draw a clear practical map. Post-training quantization—INT4 weight-only and FP8 weight–activation, often combined with vision-tower and vocabulary removal—is the dependable recipe: it compresses the Gemma-3-12B base to about a quarter of its on-disk size and yields up to an order-of-magnitude throughput gain while remaining within quality-estimation noise of the uncompressed model. Structural and low-rank methods (SVD, diagonal/binary approximation, heavy FFN pruning) remain risky at this scale: without stronger recovery they trade away quality that light fine-tuning does not restore. In the unconstrained track, treating model selection as a compression decision pays off—a compressed, newer sub-20B base can beat the compressed older base outright. Finally, speed is a property of the serving stack as much as of the precision, and reference-free metrics can be gamed. Future editions should freeze peak-VRAM instrumentation, add human/LLM-judge quality once references are released, and report speed with model-load time separated from steady-state decoding.

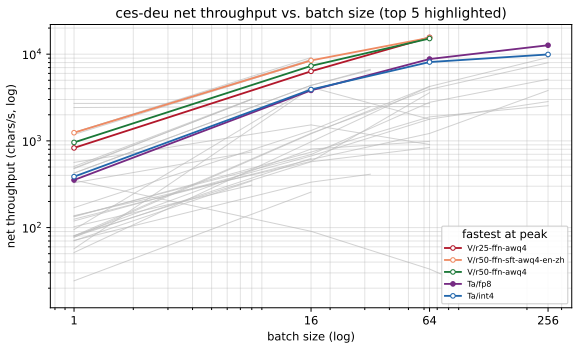



Figure 2: Net throughput vs. batch size (Czech→German, log–log). The five systems with the highest peak rate are highlighted; the rest are drawn in grey.

## 6 Limitations

Several caveats apply to this year's evaluation. First, the WMT26 test sets are blind at the time of writing, so all quality numbers here are reference-free quality-estimation scores; reference-based COMET/MetricX and the LLM-as-a-judge protocol are deferred until references are released, and QE metrics are known to be gameable (§4.1) and noisier at paragraph granularity. Second, the footprint dimension reported here is the on-disk artifact size; peak GPU VRAM is not yet instrumented uniformly across the heterogeneous runtimes, and the recorded maximum host resident-set size is not comparable across serving stacks, so we defer a VRAM comparison. Third, throughput is measured end-to-end (including model load) on a modest workload, which favors fast-loading runtimes; single-stream batch-1 latency is available only for the Czech→German speed subset. Fourth, all measurements come from a single H100 con-

[4]We suspect some submissions may not have strictly enforced batch size as a strict parallelism limiter due to the use of continuous batching in the underlying serving runtimes such as vLLM.

| System | net chars/s at batch 1 | 16 | 64 | 256 | peak | @ b |
|---|---|---|---|---|---|---|
| V/r25-ffn-awq4 | 830 | 6 343 | 15 620 | – | 15 620 | 64 |
| V/r50-ffn-sft-awq4-en-zh | 1 242 | 8 465 | 15 450 | – | 15 450 | 64 |
| V/r50-ffn-awq4 | 959 | 7 318 | 15 150 | – | 15 150 | 64 |
| **Ta/fp8** | 354 | 3 822 | 8 785 | 12 670 | 12 670 | 256 |
| Ta/int4 | 387 | 3 911 | 8 110 | 9 923 | 9 923 | 256 |
| **F/sq-gptq** | 77 | 1 206 | 4 239 | 9 160 | 9 160 | 256 |
| V/r50-ffn-sft-awq4-en-ar | 1 247 | 9 134 | – | – | 9 134 | 16 |
| V/r50-ffn-sft-awq4-cs-de | 1 253 | 8 681 | – | – | 8 681 | 16 |
| **V/r50-ffn-sft-awq4** | 1 186 | 8 399 | – | – | 8 399 | 16 |
| **F/gptq-rtn** | 80 | 1 224 | 3 927 | 7 963 | 7 963 | 256 |
| **C/gptq** | 425 | 4 364 | – | – | 6 690 | 32 |
| C/awq | 424 | 4 367 | – | – | 6 526 | 32 |
| Ti/8b-distill | 135 | 696 | 2 784 | 5 134 | 5 134 | 256 |
| **P/int4-gptq** | 52 | – | 4 244 | – | 4 244 | 64 |
| **P/gemma4-int4** | 95 | 4 146 | 1 778 | – | 4 146 | 16 |
| P/pruned4-int4 | 57 | 4 051 | – | – | 4 051 | 16 |
| Ti/8b-distill-fp8 | 125 | 611 | 1 218 | 3 812 | 3 812 | 256 |
| Ti/8b-distill-nvfp4 | 137 | 580 | 3 592 | – | 3 592 | 64 |
| **Ar/vocaball-int4** | 507 | – | – | – | 3 046 | 8 |
| Ar/fp8 | 477 | – | – | – | 3 023 | 8 |
| **Ar/int4** | 486 | – | – | – | 3 012 | 8 |
| Ti/15b-fp8 | 119 | 669 | 1 780 | 2 840 | 2 840 | 256 |
| Ta/fp8-bok4 | 2 704 | 2 700 | 2 694 | – | 2 704 | 1 |
| **Ti/15b-nvfp4** | 133 | 742 | 1 891 | 2 545 | 2 545 | 256 |
| **Ta/int4-bok4** | 2 424 | 2 500 | 2 478 | – | 2 500 | 16 |
| TM/diba-cached | 170 | 1 315 | – | – | 2 483 | 32 |
| **Tn/svd-q512-f1620-gptq4** | 564 | 1 529 | 907 | – | 1 529 | 16 |
| **Tn/gptq-int4** | 80 | 805 | 986 | – | 986 | 64 |
| B/bnb-q4 | 81 | 574 | 838 | – | 838 | 64 |
| Ar/en-ar-mbr | 328 | – | – | – | 745 | 8 |
| B/base | 78 | 626 | – | – | 626 | 16 |
| **TM/diba-triton** | 71 | 334 | – | – | 413 | 32 |
| **Tn/svd-q512-f1620** | 352 | 91 | 33 | – | 352 | 1 |
| **Al/mlp-nf4** | 70 | – | – | – | 346 | 8 |
| **S/pruned-nf4** | 102 | – | – | – | 309 | 8 |
| B/bnb-q8 | 24 | 257 | – | – | 257 | 16 |
| **C/mixed** | – | – | – | – | – | – |

Table 3: Detailed Czech→German inference speed: net throughput (characters per second, model-load time subtracted) at each batch size, plus the peak reliable rate and the batch that achieves it. "–" marks batches a system did not run or where load dominated the wall time ($> 65\%$), making the net estimate unreliable. Participant IDs follow Table 1; primary submissions are bold. Ordered by peak rate.

figuration, and the unconstrained track mixes heterogeneous base models, so its systems are only loosely comparable and are benchmarked against the constrained base as a reference rather than on equal footing.

## References


Idris Abdulmumin, Victor Agostinelli, Tanel Alumäe, Antonios Anastasopoulos, Luisa Bentivogli, Ondřej Bojar, Claudia Borg, Fethi Bougares, Roldano Cattoni, Mauro Cettolo, Lizhong Chen, William Chen, Raj Dabre, Yannick Estève, Marcello Federico, Mark Fishel, Marco Gaido, Dávid Javorský, Marek Kasztelnik, and 33 others. 2025. Findings of the IWSLT 2025 evaluation campaign. In *Proceedings of the 22nd International Conference on Spoken Language Translation (IWSLT 2025)*, pages 412–481, Vienna, Austria (in-person and online). Association for Computational Linguistics.

David Ifeoluwa Adelani, Victor Agostinelli, Antonios Anastasopoulos, Luisa Bentivogli, Ondřej Bojar, Sébastien Bratières, Marine Carpuat, Fabrício Carraro, Roldano Cattoni, Mauro Cettolo, Lizhong Chen, Marcello Federico, Marco Gaido, Mahendra Gupta, HyoJung Han, Ali Hatami, Lewis C. Howe, Dávid Javorský, Yejin Jeon, and 36 others. 2026. Speech translation and metrics in 2026: Findings of the IWSLT campaign. In *Proceedings of the 23rd International Conference on Spoken Language Translation (IWSLT 2026)*, pages 336–422, San Diego, USA (in-person and online). Association for Computational Linguistics.

Alexandra Birch, Andrew Finch, Minh-Thang Luong, Graham Neubig, and Yusuke Oda. 2018. Findings of the second workshop on neural machine transla-

tion and generation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 1–10, Melbourne, Australia. Association for Computational Linguistics.

Marco Gaido, Roman Grundkiewicz, Thamme Gowda, and Matteo Negri. 2025. Findings of the WMT 2025 shared task on model compression: Early insights on compressing LLMs for machine translation. In *Proceedings of the Tenth Conference on Machine Translation*, pages 484–494, Suzhou, China. Association for Computational Linguistics.

Hiroaki Hayashi, Yusuke Oda, Alexandra Birch, Ioannis Konstas, Andrew Finch, Minh-Thang Luong, Graham Neubig, and Katsuhito Sudoh. 2019. Findings of the third workshop on neural generation and translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 1–14, Hong Kong. Association for Computational Linguistics.

Kenneth Heafield, Hiroaki Hayashi, Yusuke Oda, Ioannis Konstas, Andrew Finch, Graham Neubig, Xian Li, and Alexandra Birch. 2020. Findings of the fourth workshop on neural generation and translation. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 1–9, Online. Association for Computational Linguistics.

Kenneth Heafield, Biao Zhang, Graeme Nail, Jelmer Van Der Linde, and Nikolay Bogoychev. 2022. Findings of the WMT 2022 shared task on efficient translation. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 100–108, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Kenneth Heafield, Qianqian Zhu, and Roman Grundkiewicz. 2021. Findings of the WMT 2021 shared task on efficient translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 639–651, Online. Association for Computational Linguistics.

Juraj Juraska, Daniel Deutsch, Mara Finkelstein, and Markus Freitag. 2024. MetricX-24: The Google submission to the WMT 2024 metrics shared task. In *Proceedings of the Ninth Conference on Machine Translation*, pages 492–504, Miami, Florida, USA. Association for Computational Linguistics.

Ricardo Rei, Nuno M. Guerreiro, JosÃ© Pombal, Daan van Stigt, Marcos Treviso, Luisa Coheur, José G. C. de Souza, and André Martins. 2023. Scaling up CometKiwi: Unbabel-IST 2023 submission for the quality estimation shared task. In *Proceedings of the Eighth Conference on Machine Translation*, pages 841–848, Singapore. Association for Computational Linguistics.

Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. 2024. A survey on model compression for large language models. *Transactions of the Association for Computational Linguistics*, 12:1556–1577.